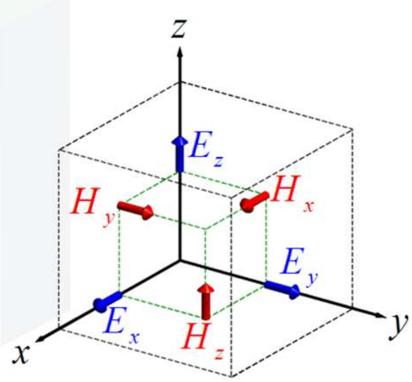
Computational Science:
Introduction to Finite-Difference Time-Domain

Introduction

Lecture Outline

- What is FDTD?
- Theory of FDTD
- How FDTD will be implemented in this class.



What is FDTD?

What is FDTD?

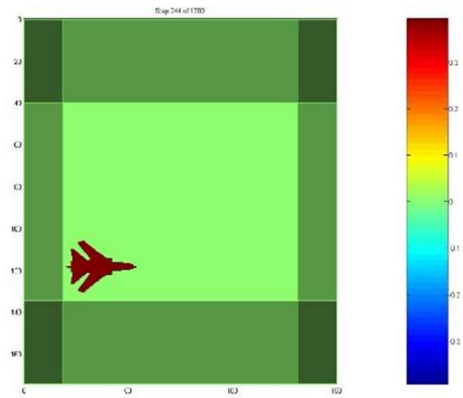
Finite-difference time-domain (FDTD) solves the electromagnetic wave equation in the time domain using finite-difference approximations. Being a time-domain method, FDTD is more intuitive than other techniques and works by creating a "movie" of the fields flowing through a device.

W. Saj, "FDTD simulations of 2D plasmon waveguide on silver nanorods in hexagonal lattice," Opt. Express **13**, 4818-4827 (2005) .

EMPossible

Slide 4

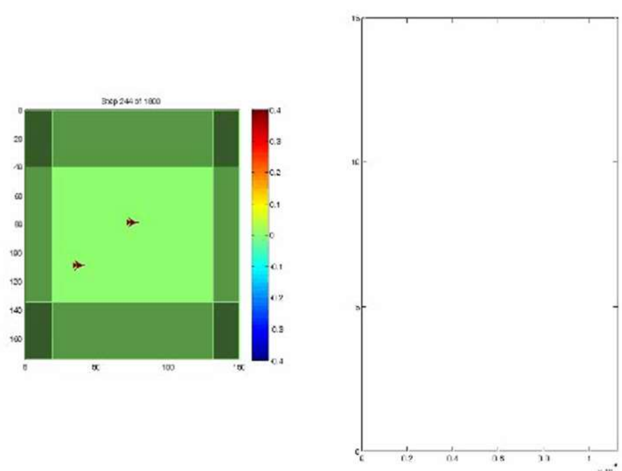
Simulation Example -- RADAR



EMPossible

Slide 5

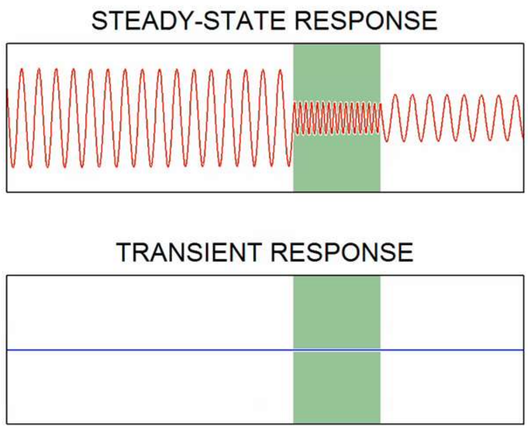
Simulation Example -- RADAR



EMPossible

Slide 6

Time-Domain Vs. Frequency-Domain



Frequency-Domain Solution

Here we only see snapshot of the field after a “long” time has passed.

We cannot directly observe the transient response.

Time-Domain Solution

Here we can watch the fields evolve over time.

We can observe the transient response.

EMPossible

Theory of FDTD

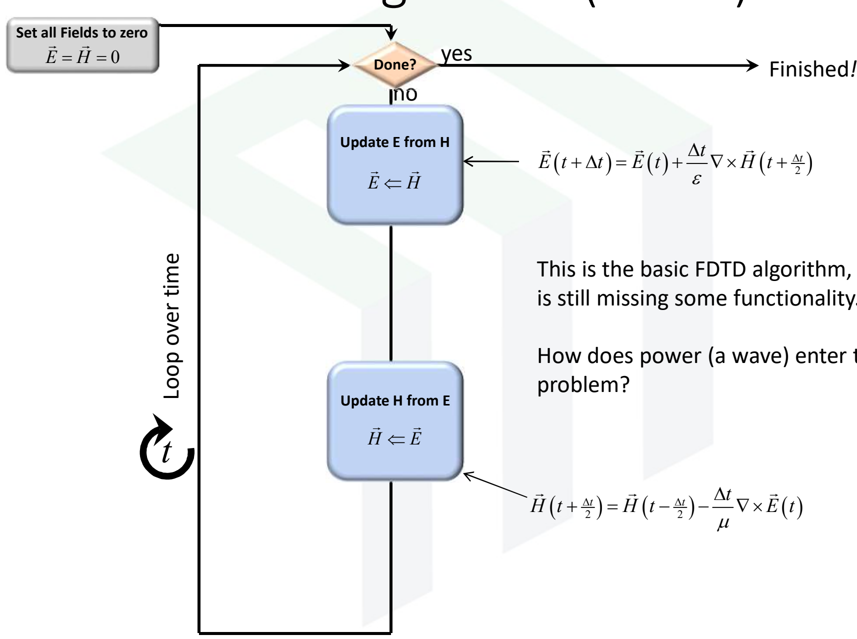
The Math Behind FDTD

$\nabla \times \vec{E} = -\mu \frac{\partial \vec{H}}{\partial t}$	Maxwell's Equations	$\nabla \times \vec{H} = \varepsilon \frac{\partial \vec{E}}{\partial t}$
$\downarrow$		$\downarrow$
$\frac{\partial \vec{H}}{\partial t} = -\frac{1}{\mu} \nabla \times \vec{E}$	Rearrange Terms	$\frac{\partial \vec{E}}{\partial t} = \frac{1}{\varepsilon} \nabla \times \vec{H}$
$\downarrow$		$\downarrow$
$\frac{\vec{H}\left(t+\frac{\Delta t}{2}\right) - \vec{H}\left(t-\frac{\Delta t}{2}\right)}{\Delta t} = -\frac{1}{\mu} \nabla \times \vec{E}(t)$	Finite-Difference Approximation	$\frac{\vec{E}(t+\Delta t) - \vec{E}(t)}{\Delta t} = \frac{1}{\varepsilon} \nabla \times \vec{H}\left(t+\frac{\Delta t}{2}\right)$
$\downarrow$		$\downarrow$
$\vec{H}\left(t+\frac{\Delta t}{2}\right) = \vec{H}\left(t-\frac{\Delta t}{2}\right) - \frac{\Delta t}{\mu} \nabla \times \vec{E}(t)$	Update Equation	$\vec{E}(t+\Delta t) = \vec{E}(t) + \frac{\Delta t}{\varepsilon} \nabla \times \vec{H}\left(t+\frac{\Delta t}{2}\right)$

EMPossible

Slide 9

FDTD Algorithm (1 of 6)



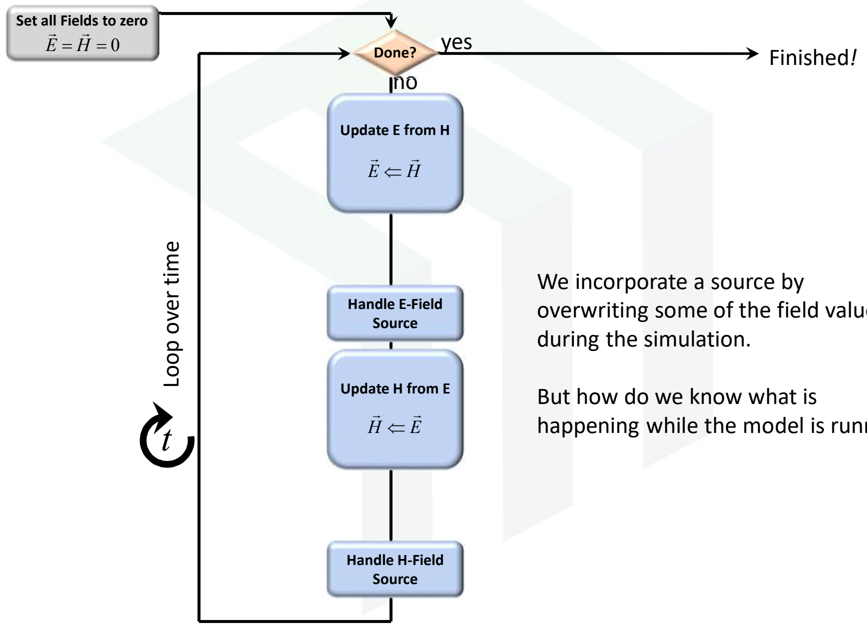
This is the basic FDTD algorithm, but it is still missing some functionality.

How does power (a wave) enter the problem?

EMPossible

Slide 10

FDTD Algorithm (2 of 6)



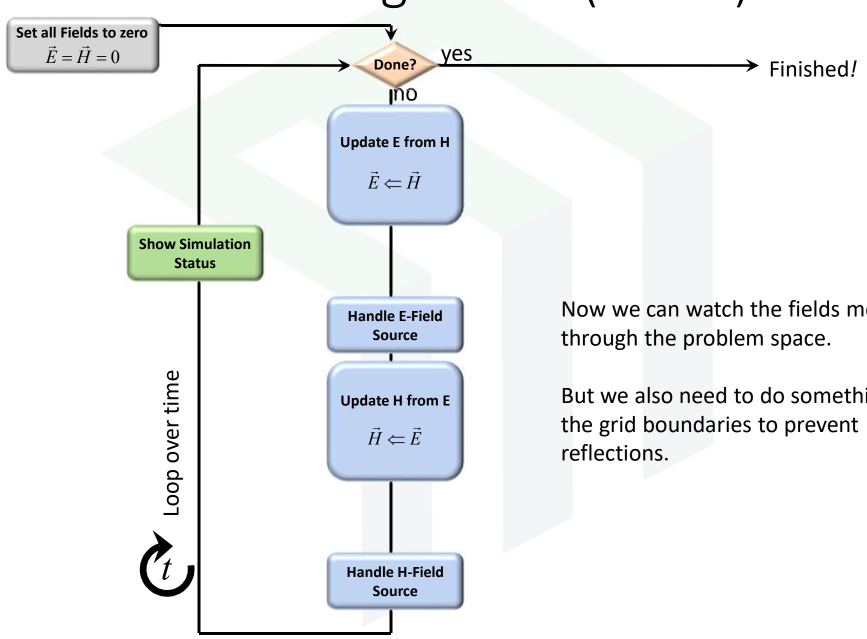
We incorporate a source by overwriting some of the field values during the simulation.

But how do we know what is happening while the model is running?

EMPossible

Slide 11

FDTD Algorithm (3 of 6)



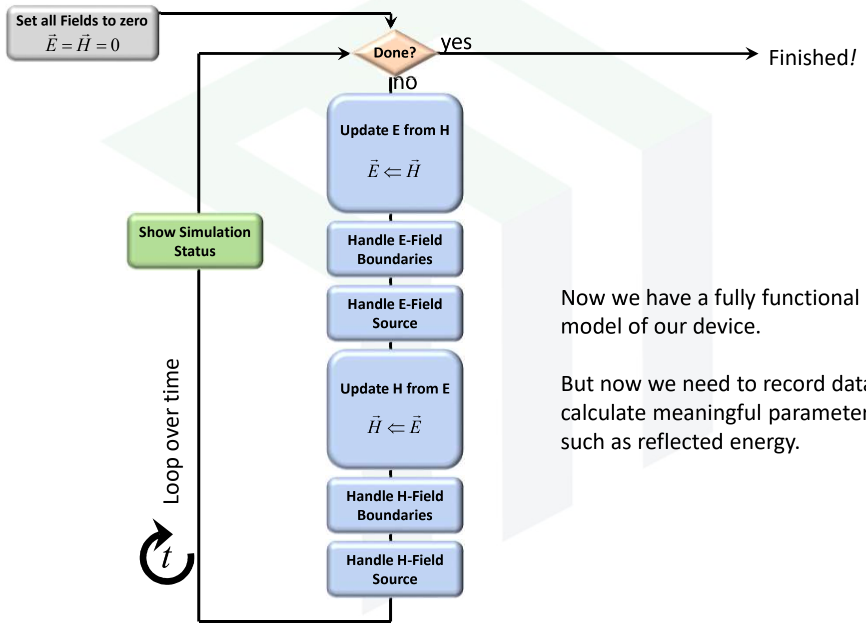
Now we can watch the fields move through the problem space.

But we also need to do something at the grid boundaries to prevent reflections.

EMPossible

Slide 12

FDTD Algorithm (4 of 6)



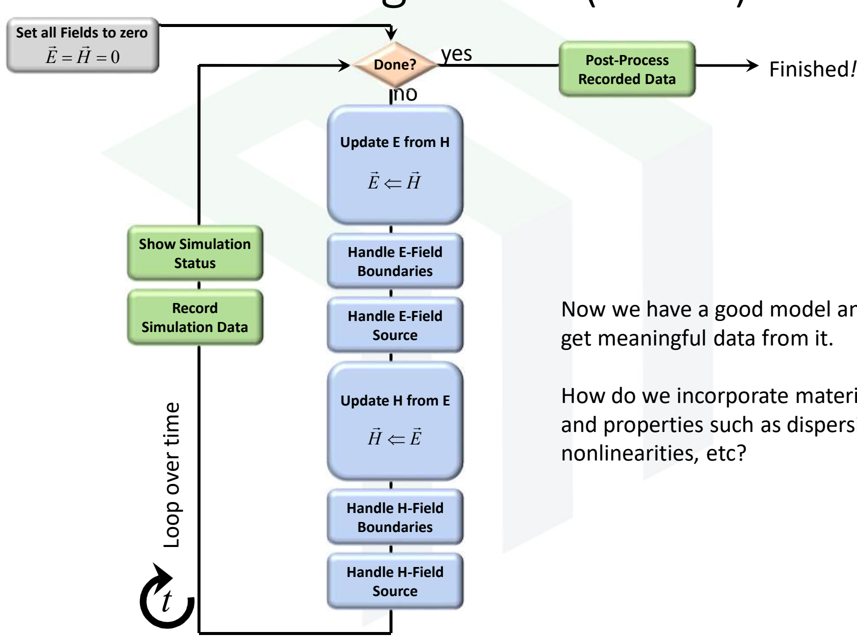
Now we have a fully functional FDTD model of our device.

But now we need to record data and calculate meaningful parameters such as reflected energy.

EMPossible

Slide 13

FDTD Algorithm (5 of 6)



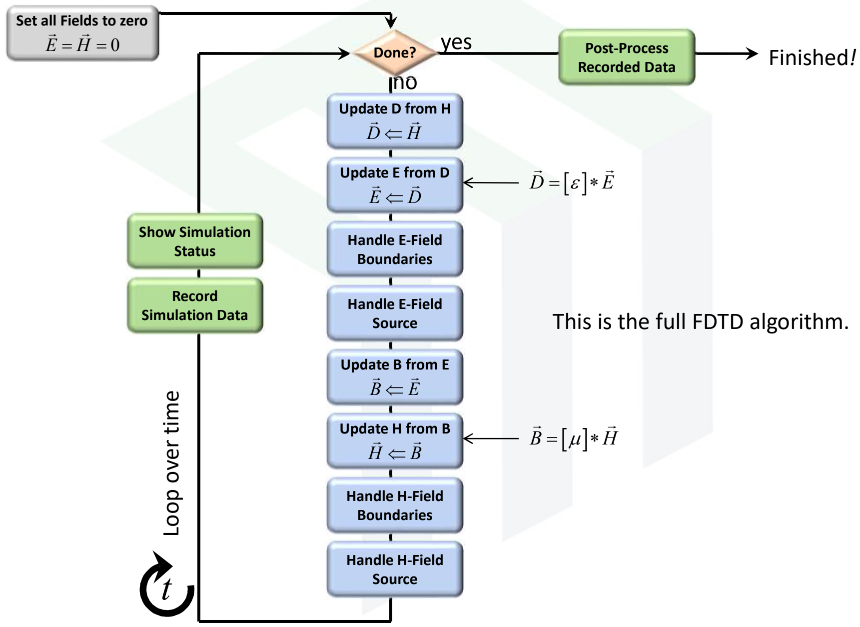
Now we have a good model and can get meaningful data from it.

How do we incorporate materials and properties such as dispersion, nonlinearities, etc?

EMPossible

Slide 14

FDTD Algorithm (6 of 6)



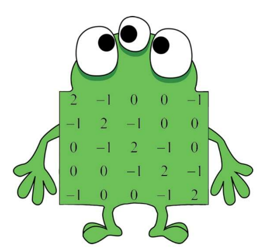
This is the full FDTD algorithm.

EMPossible

Slide 15

Benefits of FDTD

- Excellent for large scale simulations
  - It is said that numerical complexity scales linearly with problem size Typically, methods scale exponentially
  - Easily parallelized
- Excellent for broadband and/or transient simulations
- Accurate, robust, and mature method
  - Sources of error are well understood and it is a proven method in many fields
  - Lots of literature available
- Naturally handles nonlinear behavior
  - Directly handles nonlinearities due to nonlinear materials or incorporation of circuit elements
- Great for learning electromagnetics
  - Field animations and direct simulation of Maxwell's equations make FDTD a great learning tool
- **When all else fails, FDTD does not!**



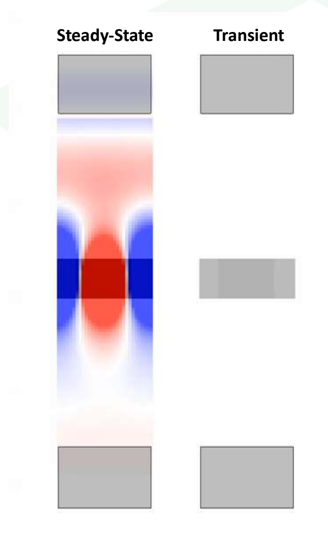
EMPossible

Slide 16

Drawbacks of FDTD

- Tedious to incorporate dispersion
- Tedious to incorporate PML
- Structured grid does not efficiently represent curved surfaces
- Slow for small devices
- Very inefficient for highly resonant devices

Time-Domain Simulation of Highly Resonant Devices



For More Information on FDTD

- http://en.wikipedia.org/wiki/Finite-difference_time-domain_method
- http://www.fdtd.org/
- For a simpler introduction
  - Dennis M. Sullivan, *Electromagnetic Simulation Using the FDTD Method*, IEEE Press.
- For more advanced topics
  - Allen Taflove, et al, *Advances in FDTD Computational Electrodynamics*, Artech House.

EMPossible

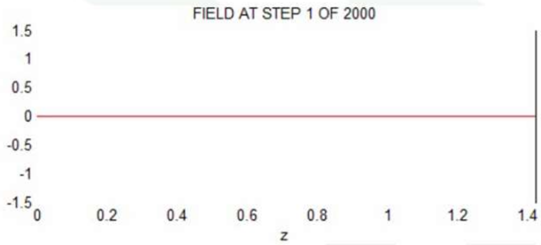
Slide 19

How FDTD Will Be Implemented in This Class

Slide 20

Step 1 – Basic 1D-FDTD Algorithm

- Basic update equations

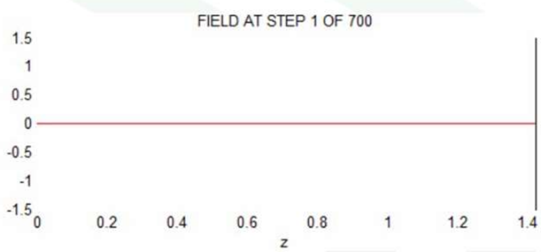


EMPossible

Slide 21

Step 2 – Add Simple Soft Source

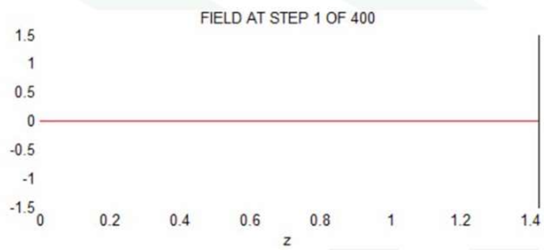
- Basic update equations
- Add a soft source



EMPossible

Slide 22

Step 3 – Add Absorbing Boundary

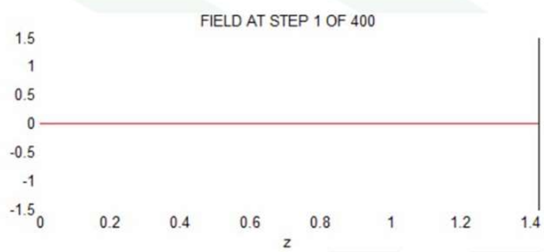


- Basic update equations
- Add a soft source
- Add perfect boundary condition

EMPossible

Slide 23

Step 4 – Add TF/SF

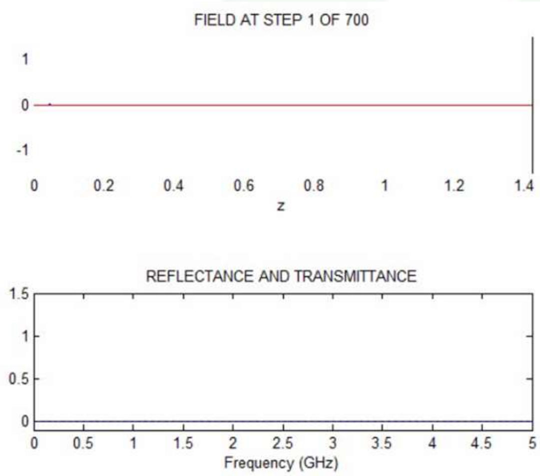


- Basic update equations
- Add a soft source
- Add perfect boundary condition
- Incorporate TF/SF "one-way" source

EMPossible

Slide 24

Step 5 – Move Source & Add T/R

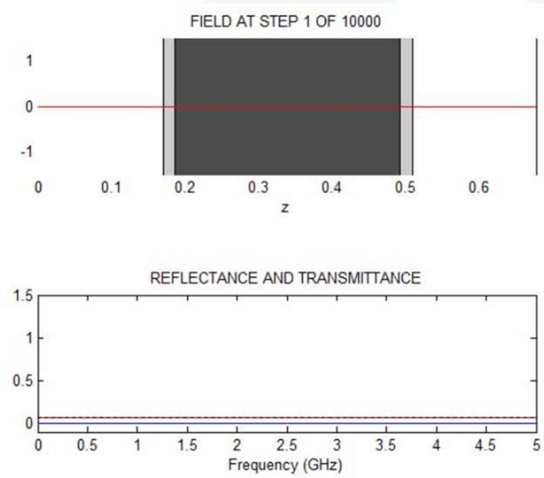


- Basic update equations
- Add a soft source
- Add perfect boundary condition
- Incorporate TF/SF "one-way" source
- Move position of source
- Calculate transmittance and reflectance

EMPossible

Slide 25

Step 6 – Add Device (Complete 1D FDTD)



- Basic update equations
- Add a soft source
- Add perfect boundary condition
- Incorporate TF/SF "one-way" source
- Move position of source
- Calculate transmittance and reflectance
- Add a real device

EMPossible

Slide 26

Step 7 – Basic 2D-FDTD Update Equations

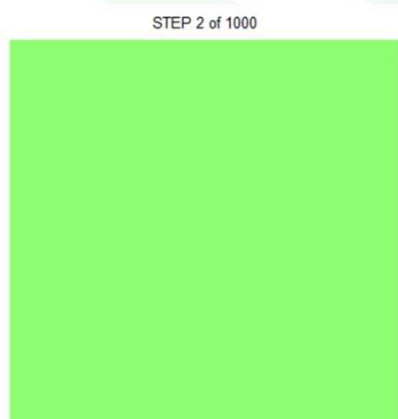
The basic update equations are implemented along with simple Dirichlet boundary conditions.

EMPossible

Slide 27

Step 8 – Incorporate Periodic Boundaries

Periodic boundary conditions are incorporated so that a wave leaving the grid reenters the grid at the other side.

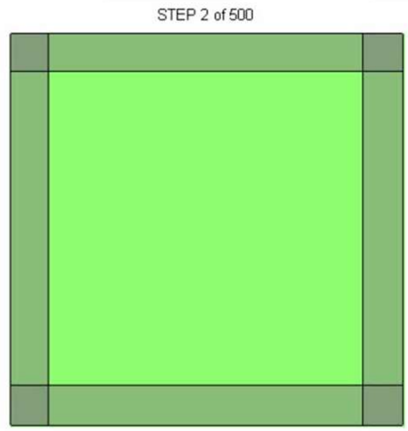


EMPossible

Slide 28

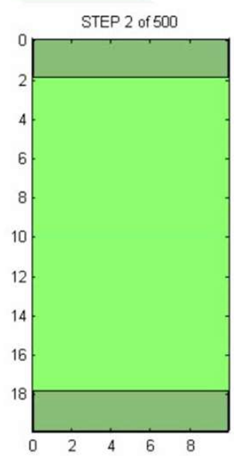
Step 9 – Incorporate a PML

The perfectly matched layer absorbing boundary condition is incorporated to absorb outgoing waves.



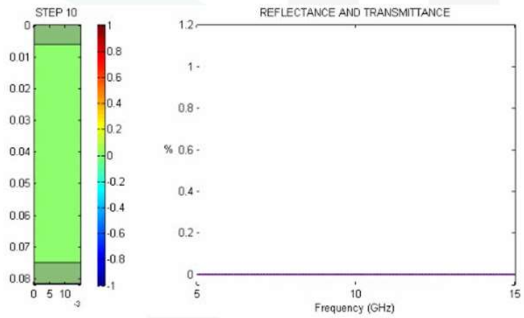
Step 10 – Total-Field/Scattered-Field

We use periodic boundaries for the horizontal axis and the PML for the vertical axis. When then implement TF/SF at the center vertically.



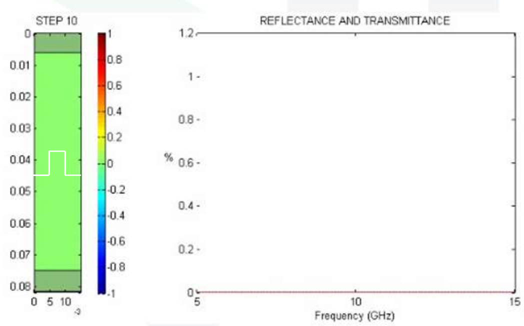
Step 11 – Calculate TRN, REF, and CON

We move the TF/SF interface to a unit cell or two outside the top PML. We include code to calculate Fourier transforms and then transmittance, reflectance, and conservation of energy.

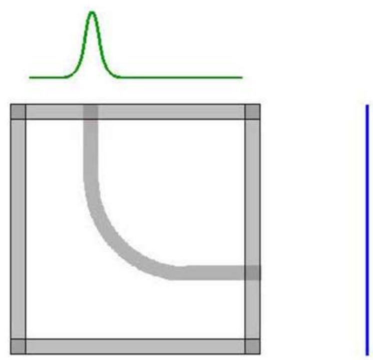


Step 12 – Add Device (Complete 2D FDTD)

We build a device on the grid that has a known solution. We run the simulation and duplicate the known results to benchmark the code.

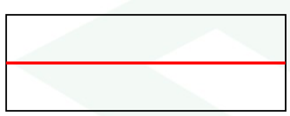


Step 13 – Waveguide Analysis

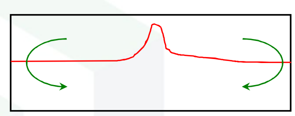


Summary of Code Development Sequence for 1D FDTD

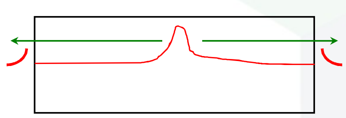
Step 1 – Implement basic FDTD algorithm



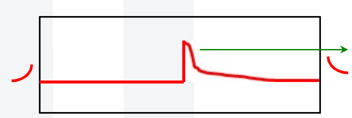
Step 2 – Add the source



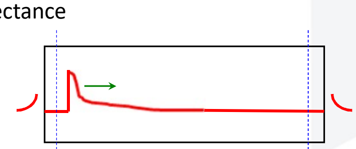
Step 3 – Add absorbing boundary



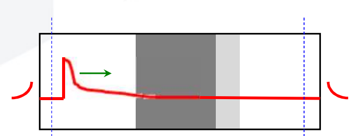
Step 4 – Add “one-way” source



Step 5 – Calculate transmittance and reflectance

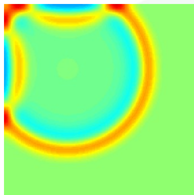


Step 6 – Add a device

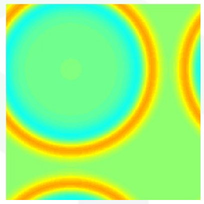


Summary of Code Development Sequence for 2D FDTD

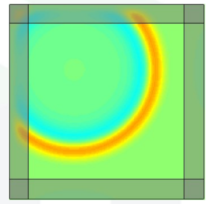
Step 7 – Basic Update + Dirichlet



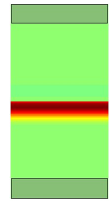
Step 8 – Basic Update + Periodic BC



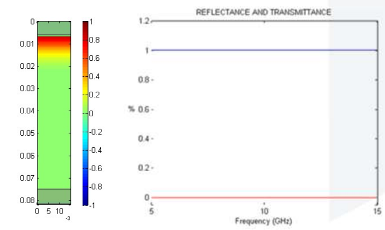
Step 9 – Add PML



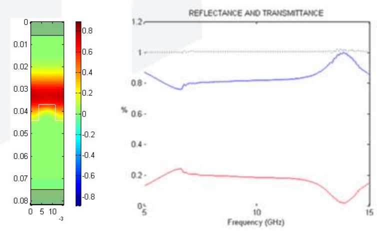
Step 10 – TF/SF



Step 11 – Calculate Response



Step 12 – Add a Device and Benchmark



EMPossible

Slide 35